Computer-Aided Design Package for Designers of Digital Optical Computers

Annual Report for Grant #N00014-90-J-4018 for Period 5/1/91 - 4/30/92

Principal Investigator: Miles Murdocca
Department of Computer Science
Rutgers University, Hill Center
New Brunswick, NJ 08903
(908) 932-2654
murdocca@cs.rutgers.edu

DTIC
ELECTE
AUG 0 5 1992
S **D**
A

May, 1992

Summary

This report covers progress for the period 5/1/91 - 4/30/92 on a jointly sponsored AFOSR/ONR grant to Rutgers University that supports research in architecture and design of digital optical computers. Progress for this reporting period includes the development of an interactive design tool for digital optical circuits, the development of optical interconnection methods, and an investigation into the architectural implications of reconfigurable optical interconnects. Prior to this reporting period, an emphasis had been placed on the Bell Labs style of digital optical processor, in which arrays of optical logic gates are interconnected in free space with regular patterns at the gate level. The computer-aided design (CAD) tools and the optical interconnection methods that we have been developing allow us to characterize tradeoffs between the complexity of the optical interconnects and the complexity of the computer architecture. We plan to spend the final year of the effort engaged in characterizing these tradeoffs, a few of which are described below in the "Micro *vs.* Macro-Optics" section. One tradeoff that is of particular interest is in how the regularity of the interconnects affect the complexity of the optics, and how the regularity affects the depth and breadth of the resulting circuits.

As an example of the tradeoff investigation that is planned for the final year, consider that irregular interconnects can be achieved with diffractive optical elements. However, studies by Stone show that there is a tradeoff between lens size and propagation distance. A completely irregular interconnect will effectively require a separate imaging system for each optical signal, and the resulting propagation distance of a few millimeters may not allow for steep angles of incidence, thus complicating the realization of a completely irregular interconnect. A mix of regular and irregular interconnects appears to be a reasonable compromise when the tradeoffs among the optics and architecture are considered together. One rule of thumb that we are exploring is to use regular interconnects for clusters of signals, 16x16 for example, and then to use irregular interconnects between clusters. In this way, propagation distance can be increased while simultaneously reducing circuit depth that is attributed to the regularity.

Interactive Design Tool for Digital Optical Circuits

Our initial work in designing digital optical circuits was based on an architecture that uses arrays of optical logic gates and regular free-space interconnects at the gate level such as perfect shuffles,
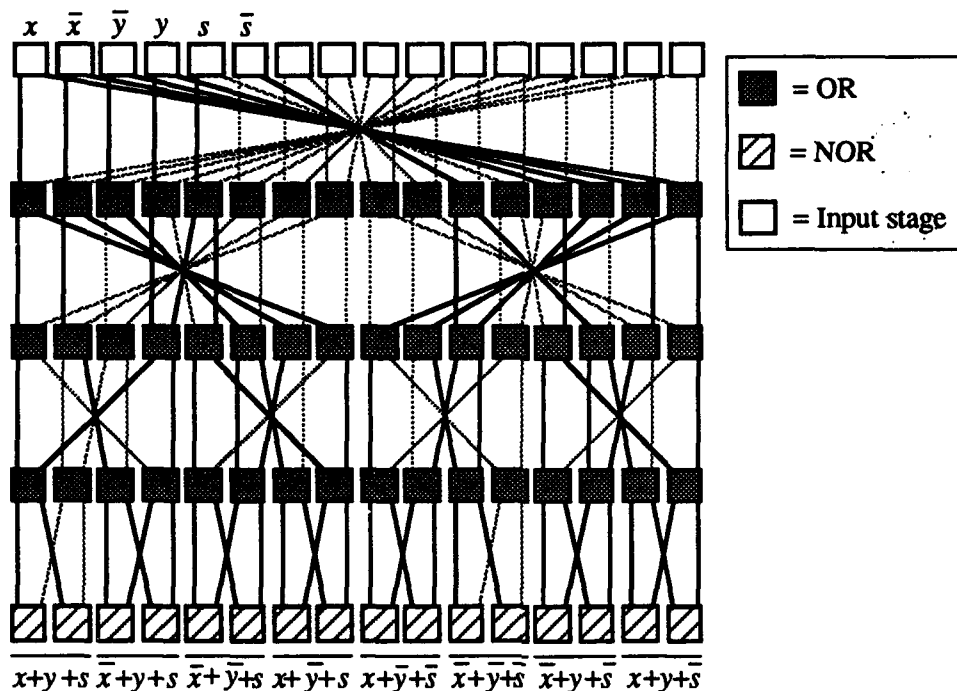
92-14744

92 6 03 144

banyars and crossovers. By suitably masking connections, Boolean functions can be implemented. Figure 1 shows a circuit that implements a 3-to-8 decoder (lightly shaded lines indicate masked connections).

$$x \quad \bar{x} \quad \bar{y} \quad y \quad s \quad \bar{s}$$

Legend:
- ■ = OR
- ◨ = NOR
- □ = Input stage

$$x+y+s \quad \bar{x}+y+s \quad \bar{x}+\bar{y}+s \quad x+\bar{y}+s \quad x+\bar{y}+\bar{s} \quad \bar{x}+\bar{y}+\bar{s} \quad \bar{x}+y+\bar{s} \quad x+y+\bar{s}$$

Figure 1: *A 3-to-8 decoder circuit is implemented by masking connections between optical logic gates. Flow of data is from the top to the bottom.*

By maintaining strict regularity at the gate level, the only flexibility left to the designer is in choosing which connections to mask, and where to place the inputs and outputs. While algorithms exist to automatically generate masks for a few basic circuits, good algorithms for generating general circuits do not exist. The circuit shown in Figure 1 is a 3-to-8 decoder which produces all eight possible combinations of the three input variables x, y, and z. A recursive algorithm for producing a general decoder is described by Murdocca *et al.* in Ref. [1]. An unreported algorithm created by Gupta for generating multiplexors also exists. However, there are many circuit design problems, such as in connecting small optical circuits to form larger circuits, for which the only known algorithms employ an exhaustive search of all possible combinations of masked and unmasked connections. For situations such as this, in which the essence of a good design cannot be captured by an algorithm, a better approach is to allow an expert to create a design interactively. Although complete automation does not currently appear practical, it does not make sense to involve the designer in aspects of design for which good automated approaches exist. Thus, the interactive design tool that we have created makes use of as much automatic layout as is currently known, and aids the designer in managing the remainder of the design.

XOPID is an interactive design tool created by Gupta, which is based on an earlier layout tool developed by Majidi. The *XOPID* tool allows logic gates to have fan-ins and fan-outs that vary, and allows circuits to have irregular interconnections between gates. These features allow us to

study the trade-offs involved when fan-in/fan-out values higher than two are used and when connections are not constrained to be regular. Other issues being studied with *XOPID* include functional decomposition and PLA tiling.

In more detail, *XOPID* is a menu-driven tool that allows the user to draw and manipulate digital optical circuits interactively in an X window. The user interface to *XOPID* is shown in Figure 2. Five vertically stacked windows comprise the display area: the command window, the file-label window, the main drawing window which contains horizontal and vertical scrollbars, the help window, and the message window. The command window contains buttons which the user selects for different circuit manipulation commands. When a button is selected, the button is highlighted, and a brief message describing its function is displayed in the help window. If the execution of a user command results in an error or some other exceptional behavior, a message is displayed in the message window. The file-label window displays the name of the circuit being manipulated.

A synopsis of the functions available to the user is given below. A more complete description is given in the user manual [2].

NEW – Clears the current circuit.

LOAD... – Prompts the user to specify a *.cir* file (a circuit file, stored in *XOPID* format). The circuit described in this file then becomes the new current circuit. If the specified file does not exist, the empty circuit becomes the new current circuit.

MERGE... – Prompts the user to specify a *.cir* file. The circuit in this file is merged into the current circuit at a position that the user can specify by clicking. The merge operation fails if a circuit overlap would be created.

SAVE – Saves the current circuit in the file named by extending the filename displayed in the file-label window with a *.cir* extension.

SAVE AS... – Prompts the user to specify a *.cir* file. The current circuit is then saved in this file.

PRINT – Prints the current circuit in PostScript format in the file named by extending the filename displayed in the file-label window with a *.ps* extension.

REFRESH – Redraws the circuit on the pixmap that is displayed in the main drawing window.

FLIP – Waits for the user to specify a rectangular region by pressing the left mouse button on the upper left corner, dragging the pointer to the lower right corner and releasing the left mouse button. A copy is made of the sub-circuit corresponding to the user-specified rectangular region, which is flipped along a vertical axis passing through the center of the region and stored in *.Clipboard.cir* from where it can be pasted using the paste option.

COPY – Similar to *FLIP* except that the sub-circuit is not flipped before it is stored in *.Clipboard.cir*.
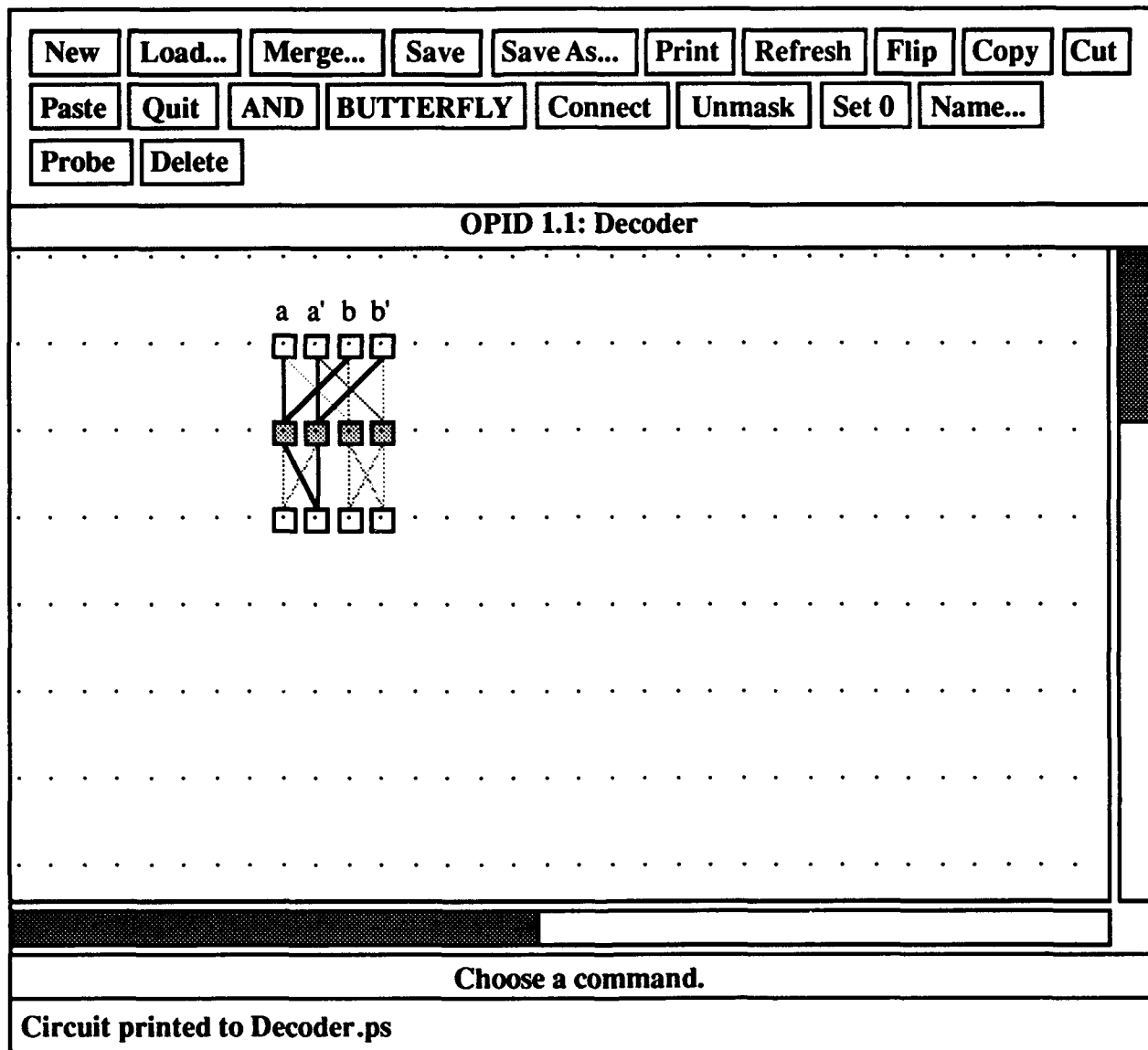
Figure 2: *The user interface to XOPID. Shaded horizontal and vertical bars serve dual functions as scrollbars and as indicators of the virtual window size.*

CUT – Similar to *COPY* but deletes the sub-circuit corresponding to the user-specified region from the current circuit.

PASTE – Waits for the user to specify a point, which is where the upper left corner of the circuit stored in *.Clipboard.cir* is merged into the current circuit, provided this does not result in an overlap.

QUIT – Exit gracefully from *XOPID*, discarding the current circuit.

OR/NOR/AND/NAND – Waits for the user to specify a rectangular region as described in *FLIP* and fills the rectangular region with gates of the type displayed in the help-window. If a gate

already exists in the region, its type is changed to that displayed in the help-window. The user can toggle through these gate types in a circular fashion by repeatedly clicking this command button.

BUTTERFLY/SHUFFLE/CROSSOVER – Waits for the user to specify a rectangular region as described in *FLIP* and inserts masked connections corresponding to the chosen pattern between gates in this region. The user can toggle through these regular interconnection patterns in a circular fashion by repeatedly clicking this command button.

CONNECT/DISCONNECT – Waits for the user to press the left mouse button over one gate, drag the pointer till it is over another gate, and release the button. If the *CONNECT* option is active, a new masked connection is made between an output of the first gate and an input of the second gate if one does not exist already. If the *DISCONNECT* option is active, the existing connection, if any, between an output of the first gate and an input of the second is removed. The active option is displayed in the help-window. The user toggles between these two options by clicking on this command button.

UNMASK/MASK – Waits for the user to specify two gates as described in *CONNECT/DISCONNECT*. If the *UNMASK* option is active, a path of connections, if one exists, leading from the output of the first gate to the input of the second gate is found and all connections on this path are unmasked. If the *MASK* option is active, all connections on the path are masked. The active option is displayed in the help-window. The user toggles between these two options by clicking on this command button.

SET 0/SET 1/UNSET – Waits for the user to click on a gate. If the *SET 0* option is active, the output of the gate that is selected is set to zero. If the *SET 1* option is active, the output of the gate that is selected is set to one. If the *UNSET* option is active, any Boolean value to which the output of the gate that is selected had been tied is removed. The active option is displayed in the help-window. The user toggles between these options by clicking on this command button.

NAME... – Prompts the user to specify a name for a gate and waits for the user to click on a gate. The output signal of the gate that is selected is then given the specified name. If a name is not specified, and if the output of the gate already has a name, that name is removed.

PROBE – Waits for the user to select a gate. The output value being generated at the gate and the Boolean expression representing its output are displayed in the message window.

DELETE – Waits for the user to specify a rectangular region as described in *FLIP*. All gates that lie inside this region are deleted from the current circuit as well as all connections that are incident on any gate inside this region.

The tools are being used as the basis of studies in tradeoffs between the complexity of the optics and the complexity of the computer architectures. Majidi is investigating scenarios in which a limited number of irregular interconnects can have a significant impact on the number of functions that can be generated in an optical digit circuit. Majidi has proven that four functions and their complements can be generated. He is working on proving the case for five functions, or if a proof is not possible, to characterize the limits on when five functions can be generated.

Optical Interconnects

Stone's work on optical interconnects is crucial in understanding tradeoffs between the complexity of the optics and the complexity of the architecture. These tradeoffs guide the directions we take in developing the CAD tools. In a separate Rome Laboratory sponsored effort, we are collaborating with the Photonics Center at Griffiss AFB in the design and construction of an all-optical digital processor based on quantum well devices. A significant problem for the RL project is in how to implement the interconnects. The calcite approach described below is one method developed by Stone that has been transitioned to the Photonics Center. The method has influenced the types of interconnects that we support in the tools, for example, the calcite approach is ideal for a split-and-shift topology.

Birefringent Array Generation and Interconnection

A hardware solution to two related problems has been demonstrated: (1) the generation of arrays of spots from a single source or from multiple sources, and (2) the interconnection of optical logic gates. Spot-array generation is a significant problem in providing power beams to modulator devices like the S-SEEDs, which are used in optical processors under development at AT&T, Boeing Aerospace, and the Photonics Center.

Cascaded slabs of birefringent materials can be used for efficient spot-array generation and for providing fan-out in optical interconnection. This was first shown, for the case of cascaded Wollaston Prisms, by Jewell [3]. In the approach studied here, collimated or converging beams are repeatedly split by propagation through simple slabs of birefringent media. These media can include, for example, calcite, rutile, quartz, or form-birefringent materials [4]. In the first stage shown in Figure 3, a spot of light polarized at 45° to the axes is imaged through a uniaxial crystal slab which is oriented with its reference plane (a plane containing both the ordinary and extra-ordinary rays) parallel to an axis. The output image of the single input spot is now resolved into two spots which are orthogonally polarized. The ordinary spot is not displaced and is polarized perpendicularly to the reference plane. The extra-ordinary spot is polarized parallel to the reference plane and is displaced by a distance proportional to the thickness of the crystal slab. In the second stage the process is repeated. Since the input spots are now polarized along the axes, the crystal slab is rotated by 45° so that each input spot retains equal components of ordinary and extra-ordinary light. The output image now consists of four spots, with orthogonal polarizations as shown. With each subsequent stage, the crystal is rotated by 45° and the number of spots are doubled. In practice, all the crystals may be optically contacted or cemented to reduce surface reflections and scatter, and a single imaging stage can be used for the entire cascade. For the case of calcite, ordinary and extraordinary rays are internally separated by an angle of 6.2°, and the crystal slabs are thus about 10 times thicker than the spot separation desired for each stage. Thin birefringent slabs with large lateral dimensions may be readily cleaved from inexpensive crystals such as calcite. The lateral extents of such slabs are not limited as with crystal prism approaches using Rochon, Wollaston, or related prisms. Other practical advantages of the cascaded slab approach include compactness, ease of manufacture, and integrability. The method has been demonstrated, and was presented at the annual OSA meeting (see Publications and Presentations section). The technology has been transitioned to the Photonics Center at Rome Laboratory where it is being considered for interconnection and for spot-array generation for their S-SEED based processor.

6
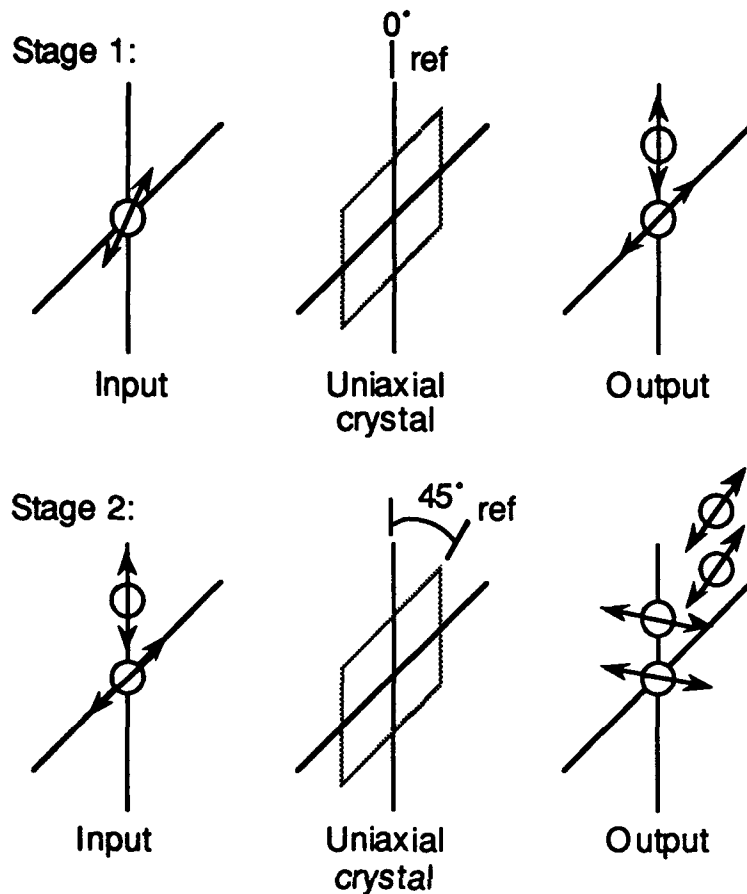
Figure 3: *Birefringent array generation.*

Sub-Array Generation, Interconnection, and Redundancy

The birefringent slab technique may be particularly useful for sub-array generation in which a sparse regular array of beams is transformed into a much denser spot array with approximately the same lateral extent. For example, an array of surface emitting microlasers may be fabricated with relatively wide element spacings to facilitate cooling. Such a coarse array can produce a dense array of spots by imaging the array through a few cascaded birefringent slabs. This is shown in Figure 4 which is a digitized photograph of an output array consisting of 48 spots. In this experiment, 12 input beams at a wavelength of 0.85µm are derived from three diode lasers and beamsplitters. These 12 beams are aligned into a regular array (simulating the output of an array of microlasers) and are focused to form spots through two cascaded calcite slabs. The two calcite slabs quadruple the density of the resulting spot array, causing each input beam to produce a local cluster of four spots. The 12 input beams and local spot clusters are shown in the diagram. The filled circles that overlay 12 of the 48 spots indicate positions in the source array. The vertices of each overlaid parallelogram indicate positions of spots that are generated from the corresponding source. This experiment also demonstrates the application of birefringent slabs to provide fan-out for optical interconnections. The light from each input beam in this example is now equally divided among four locations. Similarly, the opposite case of fan-in can be accomplished in which crystals are used to overlay light from neighboring spots. In addition to being highly efficient, birefringent array generation and interconnection are much less dispersive than diffractive

7

techniques (like Dammann gratings, which are used at AT&T in their S-SEED processors) and are therefore useful with multiple wavelengths, broadband light, or in situations where wavelengths may drift.
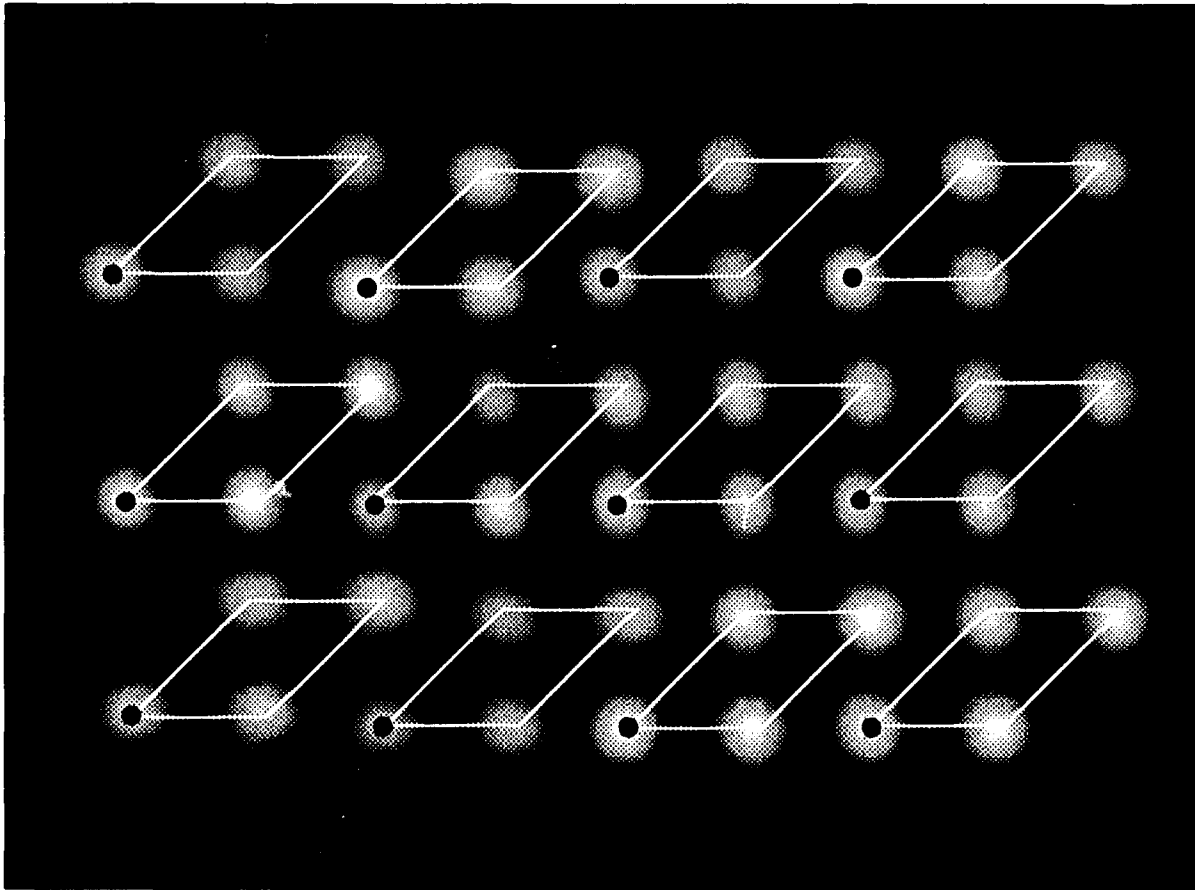
Figure 4: *Sub-array generation.*

Another important use of cascaded birefringent slabs is as an efficient method to establish redundancy in spot arrays. It was shown by Lohmann [5] that spot homogeneity and system reliability can be greatly enhanced by using spot arrays in which the spots are formed by superposing the output of a multiplicity of sources, rather than the usual case in which a single source supplies one or more spots exclusively. The reduced coherence resulting from uncorrelated source superposition greatly enhances the homogeneity of the spots by averaging out coherence related structure. Further, a large degree of source fault tolerance is achieved with this redundancy. For example, if each spot contains equal input from 32 sources, a failure of one of the input sources will only reduce the overall array uniformity by 3%. The study of using cascaded birefringent slabs for establishing such redundancy by mixing the outputs of neighboring sources in an input source array continues into the final contract year.

Progress continues on Stone's studies into how conventional and diffractive optics can be used to solve interconnect problems in computing. Stone has continued studying the characteristics of birefringent materials for spot-array generation and optical interconnection. The achromatic nature

of birefringent materials is of particular interest when compared with diffractive array generators. A paper was presented on this topic during the November 3-8 Annual Meeting of the Optical Society of America in San Jose.

Micro vs. Macro-optics
A study has continued into tradeoffs between approaches using micro-optics and macro-optics for interconnecting arrays of optical logic devices. This is an important contribution to the overall effort since it forms a basis for developing CAD tools that satisfy both fundamental and practical constraints to interconnection. Results of these analyses are reported in a book chapter that has been prepared for *Optical Computing Hardware*, edited by Sing Lee and Jürgen Jahns [6]. An excerpt from this study is given below:

Critical Distance for Collimated Array
An example of the diffraction-based trade-offs in device spacing and propagation distance is given for the case involving a collimated array of beams. The critical distance is a function of the microlens diameter. For example, consider an array with a device spacing $\Delta = 200\mu m$ and light of wavelength $0.85\mu m$. If the lens is only $10\mu m$ in diameter ($D/\Delta = 0.05$), there will be a buffer zone of width $B = 95\mu m$ on each side of the microlens over which the light may spread before crossing into the neighboring channel. The diffraction spread angle of the beam from such a small lens, however, is large (4.9°), and after only $L_c = 1.1mm$ the beam begins to spread beyond the $95\mu m$ buffer and mix with the neighboring signal. Similarly, as the microlens diameter approaches the gate spacing, the critical distance L_c is also very small. Near this other extreme, if $D/\Delta = 0.95$ ($D=190\mu m$) the diffraction angle is a much smaller .26°, but the buffer zone width is now reduced to $B=5\mu m$, and L_c is again only 1.1 mm. However, for less extreme values of D/Δ (*e.g.*, near 0.5), L_c is much larger (nearly 6mm). Figure 5 shows a plot of L_c (given in millimeters) as a function of varied fill factor D/Δ and gate spacing Δ. Since the diffraction angle decreases with increasing lens diameter, one might suspect that low crosstalk could be maintained over longer distances if the full width Δ could be utilized for the microlens apertures. Effective use of these larger apertures can be accomplished by slightly focusing the beam emerging from the microlens, thus avoiding the condition in which any spreading of the collimated beam from a lens with $D = \Delta$ results in crosstalk.

This focused array configuration is discussed in the book chapter, as well as several other results of Stone's study.

The micro/macro-optics study has been useful in identifying applications that are best served with conventional optics, and those where diffractive optics are more reasonable to apply. This aspect of Stone's work influences the design of digital circuits, and we are now using the *XOPID* tool to investigate the architectural implications of using various combinations of micro and macro-optics.

The perspective of classic lens design is an important approach to studying both conventional and diffractive optics, and to that end, the OSLO Series 2 lens design program was purchased using cost-sharing funds which were provided by Rutgers for this grant. The software is important as a research tool for studying the properties of new elements and new configurations. For example, it

9

has the flexibility for the user to introduce routines describing the characteristics of novel or unusual optical surfaces. This software was installed during this reporting period.

CRITICAL DISTANCE (mm)
COLLIMATED ARRAY
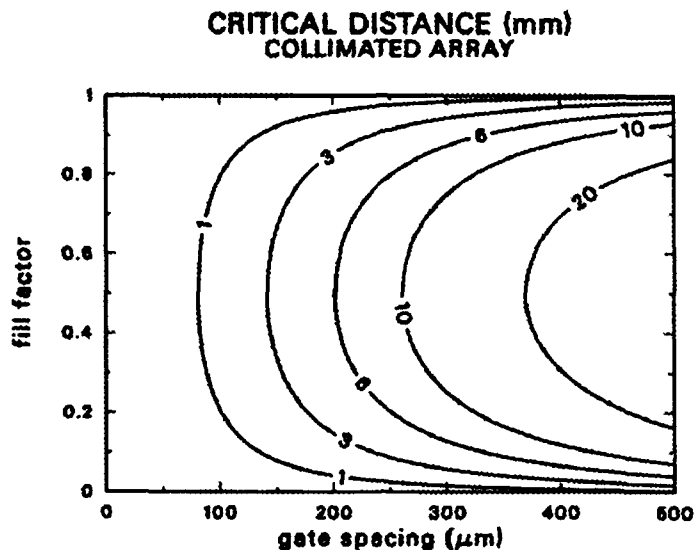
Figure 5: *Plot showing critical distance as a function of gate spacing and fill factor.*

Since the project is being carried out in the Computer Science Department at Rutgers University, not all of the computer scientists have a sufficient background in optics in order to understand the tradeoffs. A regular series of seminar meetings was organized by Stone in order to tutor the members of this research group, in which guiding principles of physical and geometrical optics were discussed along with architectural issues in computing.

Investigation into Reconfigurable Interconnects

The notion of *functional locality* was introduced by Murdocca in the final report of a Phase I SBIR project sponsored through SDIO which was administrated by AFOSR (contact is Dr. Alan Craig). The suggestion is made in that report that an ordinary computer program displays locality in terms of the kinds of instructions it executes, *i.e.* a program is likely to execute an instruction from the small set of instructions that it executed most recently. We have investigated that notion here, and have found that functional locality exists in typical computer programs.

Just as spatial and temporal locality exhibited by a program are utilized to speed up effective memory reference delays using memory caches, we believe that functional locality makes a strong case for the introduction of *function caches*. Reconfigurable processors that execute only a small set of machine instructions at any given instant but at rates faster than non-reconfigurable processors can exploit functional locality to achieve higher performance. Our belief that such a reconfigurable processor will execute its instructions faster than a comparable non-reconfigurable processor is based on the design guideline that *smaller hardware is faster* [7]. We have found evidence that functional locality exists in typical computer programs, which we report on in the remainder of this section.

10

Measurements of instruction set usage

Hennessy and Patterson [7] report on the instruction set usage for a number of application programs running on different architectures, and one of their conclusions is that programs use only a small part of the total instruction set provided by the architecture, and that an even smaller set of instructions (about twelve or so) account for as much as 80% of the total number of instructions executed. This observation motivated us to look for functional locality in programs. Our study is carried out in two parts. We look first at the extent of functional locality that arises solely from the fact that some instructions are executed more often than others. In this part of the study, using the run-time frequency count information collected in the instruction usage study reported in Ref. [7], we synthesized random runs with uniform frequency distributions of machine instructions matching those reported, and studied the hit-ratio that a reconfigurable processor would achieve for different sizes of the function cache using a first-in-first-out (FIFO) instruction replacement strategy. It should be emphasized here that it is the actual hardware that is being replaced, and not simply the codewords that represent instructions. This part of the study uses two architectures – the DEC VAX, and the DLX, which is a generic Load/Store architecture described in Ref. [7]. Three different programs are used on each machine: **gcc** (a C compiler), **spice** (a circuit simulator), and **tex** (a text formatter).

Figure 6 shows a plot of hit-ratio against code size for the DLX running **gcc** for different function cache sizes. Plots for **spice** and **tex** for the DLX, and also for these three programs on the VAX are nearly identical in form to Figure 6. For our purposes, the hit-ratio is the percentage of the total instructions executed for which reconfiguration is required assuming that at any given instant, the processor implements only as many instructions as the size of the function cache allows and reconfigures itself to implement an instruction that is not in the function cache. In each case the hit-ratio increases as the function cache size increases but is almost completely insensitive to code size. For this reason, we use a code size of no more than 1,000,000 machine instructions for the measurements that follow.

Figure 7 shows plots of hit ratios as functions of cache sizes for the DLX and the VAX, respectively, for synthesized runs based on the instruction mixes found in **gcc**, **spice**, and **tex**. As shown in the plots, high hit ratios are obtained for small cache sizes. Motivated by these results, we developed software tools for the second part of the study, which allowed us to gather entire runs of some sample programs. The architecture used for this part of the study is the SPARC based Sun-4 and the programs studied are **latex** and the **gcc** components: **gcc-cpp**, **gcc-cc1**, **as** and **ld**. Collecting a program trace in this fashion slows down the program being traced by a large factor. For example, one trace of seven million instructions required nine hours of actual time. For this reason, the programs were executed using small sample files. Figure 8 shows the effect of changing the size of the function cache on the hit-ratio for different programs. For the programs shown in Figure 8, we also gathered data on the run-time frequency distribution, generated runs with matching frequency distributions and studied the effect of function cache size on hit-ratio. Our observations are shown in Figure 9. Note that the hit-ratio values we see in Figure 8 are higher than corresponding values seen in Figure 9 for the same function cache size. This indicates that the programs in our study exhibit functional locality to a higher degree than would be exhibited simply because programs execute some instructions more often than others.
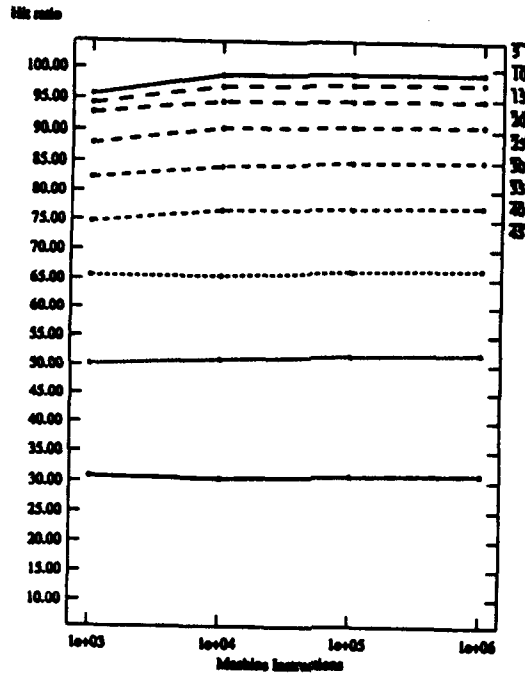
11

Figure 6: *Hit ratio versus code size for different sizes of the function cache for synthesized runs of gcc on the DLX.*
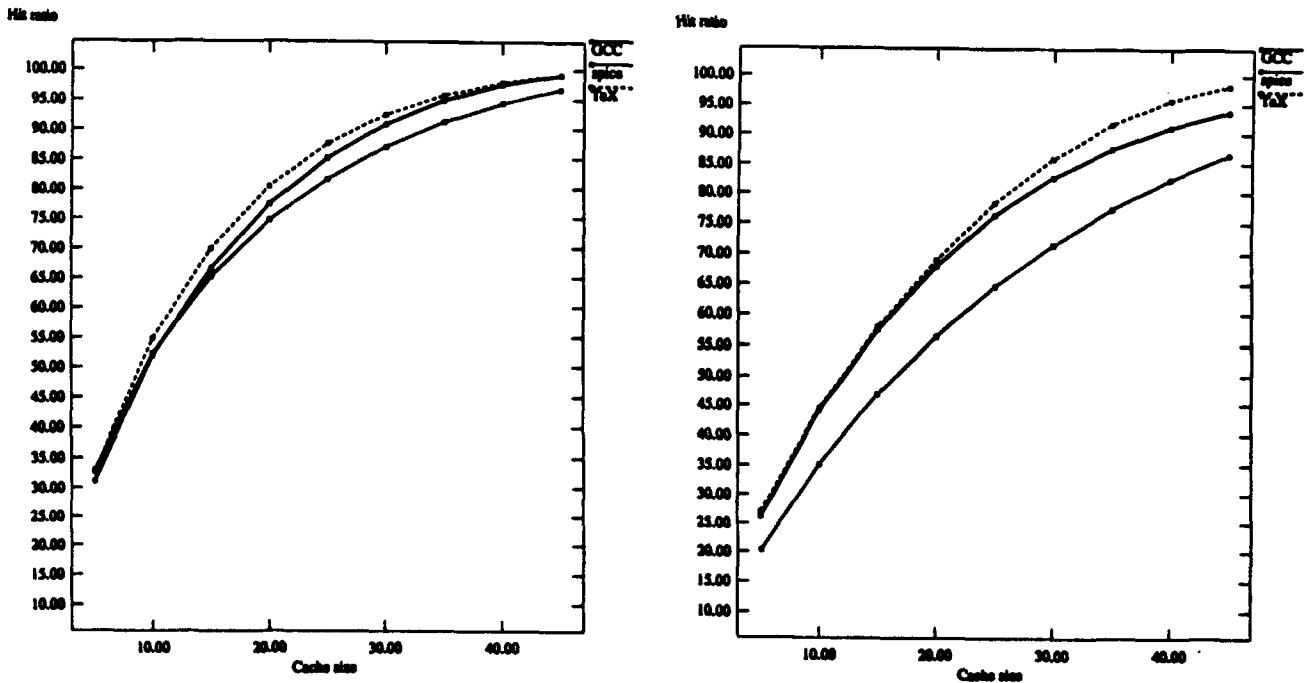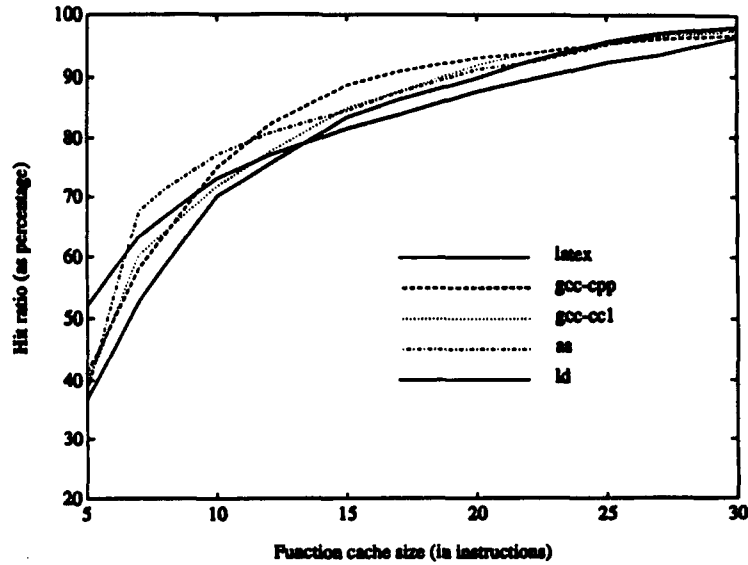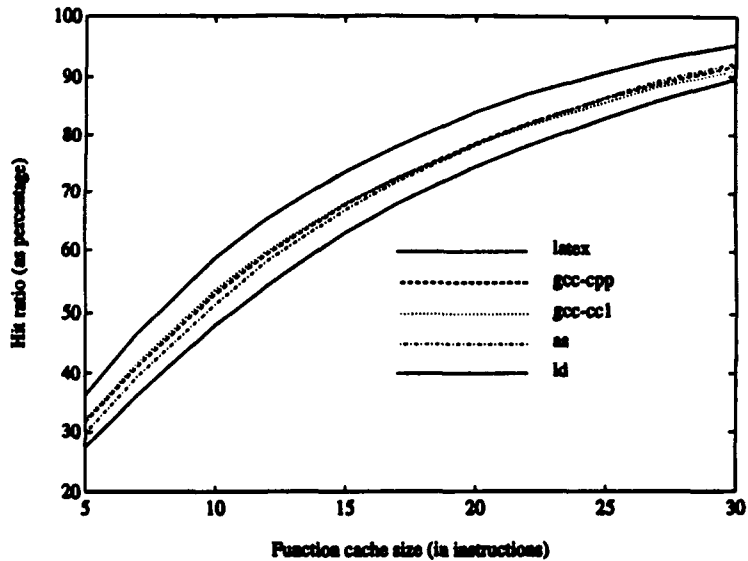

Figure 7: *Hit ratio as a function of the size of the function cache for synthesized runs on the DLX (left), and the VAX (right).*

12

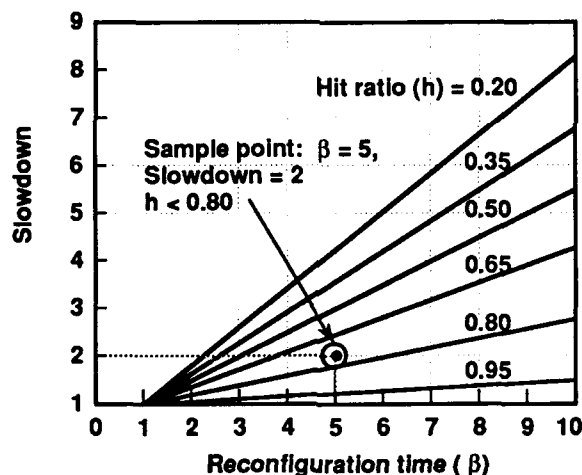**Figure 8:** *Hit ratio as a function of the size of the function cache for actual runs on the SPARC.*

Figure 9: *Hit ratio as a function of the size of the function cache for synthesized runs on the SPARC.*

Discussion

The data we have collected provides evidence for the existence of functional locality in ordinary programs. Based on the evidence, we claim that a reconfigurable processor that modifies its hardware to execute a slowly changing set of machine instructions can exploit functional locality to achieve higher performance than a non-reconfigurable processor. In order to quantify the

13

performance gain, we define β (> 1) as the ratio by which the execution of an instruction which is not in the function cache is slowed down compared to the execution of an instruction in the cache. The factor by which a reconfiguring multiprocessor is slowed down because of misses is then given by $slowdown = h + \beta \times (1 - h)$ where h is the hit-ratio. If α (< 1) is the ratio of the speed with which the reconfigurable processor executes an instruction that it finds in its cache to the speed at which a non-reconfigurable processor executes an instruction, then in order for the reconfigurable processor to be faster than the non-reconfigurable processor, α should be less than $\frac{1}{slowdown}$. From Figure 10, it is clear that the higher the hit-ratio, the lesser the sensitivity of the slowdown factor to the cost of reconfiguration. We choose a sample point $\beta = 5$, based on the expected reconfiguration time as compared to the bit rate of matrix addressable devices which are being developed at PRI under NASA support. We choose h near 0.8 which is typical for execution runs we have observed. This sample point gives us a slowdown of 2, which means that the processor runs twice as slow as a result of misses than it would run if there were no misses at all. A reconfigurable processor is assumed to be faster than a non-reconfigurable processor as a result of its reduced size, however, and so the speedup must compensate for the slowdown. The operating region in which a reconfigurable approach breaks even is shown for this sample point in Figure 11. In order for the reconfigurable processor to break even, it must execute instructions at twice the rate of a non-reconfigurable processor.

Figure 10: *Slowdown factor as a function of the reconfiguration cost and hit-ratio.*

A potentially significant application of these results is for the DOC II optical processor under development at OptiComp. The DOC II processor is being developed for a SPARC-like instruction set, but only a subset of the instruction set can be implemented at any one time. The study reported here can be used both to determine how to organize the cache (the DOC II instruction mask) and to characterize the effectiveness of the strategy.

14
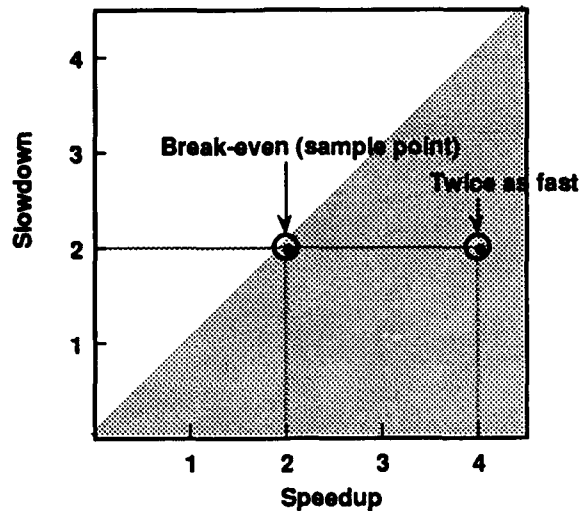
Figure 11: *Break-even operating region in which the slowdown that results from cache misses is compensated by the speedup due to the smaller size of the processor.*

Collaboration with NEC

During the past year, we have created a formal collaboration with NEC Research Institute in Princeton. Dr. Eugen Schenfeld at NEC has made arrangements to support Ph.D. student Vipul Gupta for 50% of his time, starting in July 1992. A collaboration has already been underway for several weeks, primarily between Gupta and Schenfeld, in the area of reconfigurable optical interconnects.

Gupta is investigating a model that consists of a two level, crossbar based interconnection network. This network consists of a large number of small (8x8 or 16x16) fast electronic crossbar switches at Level 1 which are connected by a very large but slow switching optical crossbar at Level 2. Processing elements connect directly to the fast switches in Level 1 and communication between processing elements attached to distinct Level 1 switches goes first through a fast switch at Level 1, then a slow switching optical network at Level 2, and then finally another fast switch at Level 1.

Multiprocessors connected in a large variety of classical topologies like trees, rings, hypercubes and meshes can be partitioned into a large number of small clusters in such a fashion that assigning each cluster to a Level 1 switch would result in a network configuration at Level 2 that does not need to be switched very often. This lets the interconnection network combine the fast switching speed of the smaller electronic switches with the high connectivity and bandwidth of a large optical network.

The main problem for this method is to partition a graph representing the communication structure of multiprocessors into clusters exhibiting the property mentioned above. A clustering algorithm based on probabilistic hill-climbing has been designed and implemented. Preliminary results on communication graphs of up to 500 nodes arranged in 2-D meshes, binary trees, ternary trees, and hypercubes are encouraging. The Level 2 switch is considered to be optical, because of the large

15

bandwidth requirement, and because switching only needs to be handled infrequently with respect to the bit rate. Gupta is continuing his investigation in this area.

References

[1] M. J. Murdocca, A. Huang, J. Jahns, and N. Streibl, "Optical Design of Programmable Logic Arrays," *Applied Optics*, **27**, pp. 1651-1660, (May 1, 1988).

[2] V. Gupta, "XOPID: An Interactive Design Tool for Digital Optical Circuits," (1992).

[3] See for example: J. L. Jewell, S. L. McCall, Y. H. Lee, A. Scherer, A. C. Gossard, and J. H. English, "Optical Computing and Related Microoptic Devices," *Appl. Opt.*, **29**, pp. 5050-5052 (1990).

[4] K. Shiraishi, T. Sato, and S. Kawakami, "Experimental Verification of a Form-Birefringent Polarization Splitter," *Appl. Phys. Lett.*, **58**, pp. 211-212, (1991).

[5] A. W. Lohmann and S. O. Sinzinger, "Spatial Noise Reduction in Array Illuminators," *OSA Optical Computing Technical Digest Series*, **6**, pp. 38-40, (1991), (Digest from Topical Meeting on Optical Computing, March 4-6, Salt Lake City, Utah, 1991).

[6] D. Smith, M. Murdocca, and T. Stone, "Parallel Optical Interconnections," book chapter in *Optical Computing Hardware*, vol. 2, edited by S. Lee and J. Jahns, Academic Press, (1992), to appear.

[7] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann Publishers, (1990).

Personnel

The AFOSR/ONR contribution amounts to 200K for this reporting period, which is in addition to 29K in cost-sharing from Rutgers University and several months of support for Murdocca. The labor breakdown for the year is summarized below. The notation X/Y represents X months work contributed to the project with Y months charged to the AFOSR/ONR grant. When the months contributed exceed the months charged, the difference is contributed by Rutgers University. CS is an abbreviation for Computer Science and EE is an abbreviation for Electrical Engineering.

Person	Time	Title	Background	Contribution
Miles Murdocca	6/0	Asst. Prof.	CS/EE	Project organization, digital circuit design.
Thomas Stone	9/9	Res. Asst. Prof.	Optical Science	Diffractive optics studies.
Vipul Gupta	12/12	Ph.D. student	CS/EE	Collision detection, circuit layout, functional locality.

16

| Masoud Majidi | 12/12 Ph.D. student | CS | Algorithms for implementing multiple functions, permutations. |

Publications and Presentations

The following publications and presentations were made during the past year. The AFOSR and ONR sponsoring agencies are acknowledged in each publication. Some of the publications name Prof. Apostolos Gerasoulis and Ph.D. student Tao Yang as authors. Although these people are not directly supported by the AFOSR/ONR grant, Rutgers University has made 97K in cost-sharing available to us in support of this grant, and we have used some of the cost-sharing resources to support their conference expenses since their work in parallel compiling supports our investigation into reconfigurable optical interconnects. Prof. Donald Smith, who appears as a co-author in the first publication, developed a method of avoiding defects in optical logic arrays which was described in the previous annual report. His support for that year was provided by Rutgers University, which is in addition to the 97K in cost-sharing.

Publications

D. Smith, M. Murdocca, and T. Stone, "Parallel Optical Interconnections," book chapter in *Optical Computing Hardware*, vol. 2, edited by S. Lee and J. Jahns, Academic Press, (1992, to appear).

M. Murdocca and V. Gupta, "Architectural Implications of Reconfigurable Optical Interconnects," submitted to: *Journal of Parallel and Distributed Computing.*

M. Murdocca and A. Sharma, "An application of optically reconfigurable interconnects to the dataflow parallel computing paradigm," *SPIE Proceedings of OE/Aerospace, Advances in Optical Information Processing V*, (Apr. 1992).

M. Murdocca "Architectural implications of optical computing," *Proceedings of the 12th GI/ITG Conference on Architecture of Computing Systems*, Kiel, Germany, (Mar. 1992).

M. Murdocca and S. Levy, "Design of a Gaussian Elimination Architecture for the DOC II Processor," *1991 OE-Lase Symposium*, (Jul. 1991).

T. Yang and A. Gerasoulis, "PYRROS: Static Task Scheduling and Code Generation for Message Passing Multiprocessors," *ACM International Conference on Supercomputing*, (July 1992, to appear).

A. Gerasoulis and T. Yang, "On the Granularity and Clustering of Directed Acyclic Task Graphs," Submitted to: *IEEE Transactions on Parallel and Distributed Systems.*

A. Gerasoulis and T. Yang, "Automatic Program Scheduling for Distributed Memory Scalable Architectures," Submitted to: *CONPAR92/VAPPV Conference,* (Sep. 1992).

A. Gerasoulis and T. Yang, "Scheduling Program Task Graphs on MIMD Architectures," to appear as a book chapter in *Algorithm Derivation and Program Transformation*, edited by R. Paige, J. Reif, and R. Wachter, Kluwer, (1992).

Presentations

T. Stone (presenter) and J. Battiato, "Spot-Array Generation and Optical Interconnection Using Birefringent Crystals," *Opt. Soc. Am. Annual Meeting*, San Jose, California, (Nov. 3-8, 1991).

T. Stone, "Quadratic Optical Pulse Compressor," *Opt. Soc. Am. Annual Meeting*, San Jose, California, (Nov. 3-8, 1991).

M. Murdocca (presenter) and A. Sharma, "An application of optically reconfigurable interconnects to the dataflow parallel computing paradigm," *SPIE OE/Aerospace,* Orlando, (Apr. 1992).

M. Murdocca, "Architectural implications of optical computing," *The 12th GI/ITG Conference on Architecture of Computing Systems*, Kiel, Germany, (Mar. 1992).

M. Murdocca (presenter) and S. Levy, "Design of a Gaussian Elimination Architecture for the DOC II Processor," *1991 OE-Lase Symposium*, San Diego, (Jul. 1991).

M. Murdocca, "Architectural Implications of Reconfigurable Optical Interconnects," presented at the Boulder Workshop on Optical Interconnects, (Feb. 1992).

A. Gerasoulis, *10th Parallel Circus*, Oak Ridge National Lab, (Oct. 1991).

A. Gerasoulis, *ACM/IEEE Supercomputing '91*, Albuquerque, New Mexico, (Nov. 1991).

A. Gerasoulis, *IEEE Scalable High Performance Computing Conference SHPCC-92*, Williamsburg, (Apr. 1992).

A. Gerasoulis, *ACM International Conference on Supercomputing (ICS92)*, Washington, (Jul. 1992).

Patent Disclosures
The following patent disclosures were made during the past year:

T. Stone, "Birefringent Array Generator and Birefringent Optical Interconnector."

T. Stone, "Achromatic and Dispersive Retarders and Compensators."

Plans for Future Work
The work planned for the final year on optical interconnects involves a continuation of the tradeoff investigation relating optical and architectural complexity, which will make use of the CAD tools. Emphasis will be on combined micro/macro approaches that retain dense element spacing while minimizing crosstalk. Diffractive and refractive lens modeling will continue, and the OSLO software is expected to be an important asset in this investigation. Birefringent interconnects,

although a comparatively simple concept, have distinct advantages including efficiency, scalability, and low dispersion. A publication on this approach will be submitted early in the final year, and work will continue in studying the impact such interconnects can have in the optical/architectural tradeoffs. The low dispersion nature of these interconnects may be very useful for interconnecting microlaser or other devices where wavelength variations can be an impediment to high dispersion diffractive interconnects. In this context other achromatization techniques for interconnects will also be studied, including hybrid diffractive/refractive elements and an evaluation of the applicability of the achromatic retarder. The laboratory facility (built from matching funds from this contract) for studying interconnects and fabricating diffractive elements will be operable during the final year and will be used for feasibility experiments.

Gupta and Murdocca plan to continue their investigation into reconfigurable interconnects, and to characterize situations in which reconfiguration can be exploited in order to improve speed or density. Any algorthms or design techniques that result from this investigation will be incorporated into the CAD tools.

An extensive software task lies ahead in making the tools more user friendly, and in refining the tools to conform to practical considerations that arise in laboratory experiments. For example, the tools are being used in a Rome Laboratory sponsored SBIR effort for the design of an all-optical Gaussian elimination processor. Some of the practical problems that have been discovered that the tools do not address are:

1) The need for a simulator that supports functional modeling and debugging of logic circuits.

2) The need to XY-fold rectangular circuit designs (this is the shape that the CAD tools produce) onto square logic arrays.

3) The need to Z-fold OR-NOR circuit designs into the linear order supported by arrays (*e.g.* OR-OR-NOR-repeat may be all that is supported when OR-NOR-OR-NOR is needed). A related issue is the need to automate the mapping of associative logic (such as AND and OR) onto non-associative logic arrays (such as NOR).

4) An implementation of Smith's fault-avoidance algorithms which were discussed in the previous Annual Report.

Although we would like to address these practical considerations at Rutgers, it poses a considerable time investment for the Ph.D. students, which will hinder their investigations in higher payoff research such as reconfigurable interconnects and better layout strategies. As an alternative, we took advantage of an opportunity that came about through the SBIR program. The tools will be upgraded through the help of two programmers who are being funded through a Phase II SBIR effort at Rome Laboratory (Project Engineer is Robert Kaminski 315-330-4092). This effort will be supervised by Murdocca to ensure a smooth transition of the tools from Rutgers to Rome Laboratory.

Conclusion

The most significant accomplishments for the period include (1) the development of a CAD tool package for digital optical circuit design, (2) the development of function caching, which exploits properties of reconfigurable optical interconnects (3) an investigation into micro vs. macro optical tradeoffs, and (4) increasing the number of functions that can be implemented on a PLA. A transfer of technology took place to Rome Laboratory for Stone's work in birefringent interconnects and spot-array generation, which resulted in one of two patent disclosures for the year.

Miles Murdocca